Real-time Facial Feature Detection using Conditional Regression Forests

Matthias Dantone[1] Juergen Gall[1,2] Gabriele Fanelli[1] Luc Van Gool[1,3]
[1]ETH Zurich, Switzerland [2]MPI for Intelligent Systems, Germany [3]KU Leuven, Belgium
{dantone,fanelli}@vision.ee.ethz.ch jgall@tue.mpg.de vangool@esat.kuleuven.be

Abstract

Although facial feature detection from 2D images is a well-studied field, there is a lack of real-time methods that estimate feature points even on low quality images. Here we propose conditional regression forest for this task. While regression forest learn the relations between facial image patches and the location of feature points from the entire set of faces, conditional regression forest learn the relations conditional to global face properties. In our experiments, we use the head pose as a global property and demonstrate that conditional regression forests outperform regression forests for facial feature detection. We have evaluated the method on the challenging Labeled Faces in the Wild [20] database where close-to-human accuracy is achieved while processing images in real-time.

Figure 1. Our approach estimates facial feature points from 2D images in real-time.

1. Introduction

Due to its relevance for many applications like human computer interaction or face analysis, facial feature point detection is a very active area in computer vision [1, 3, 24, 25, 29]. Recent state-of-the-art methods like [3] have reported impressive results where localization accuracy of human annotators has been achieved on images of medium quality. However, most of the available methods do not achieve real-time performance which is a requirement for many applications. Furthermore, low quality images still challenge state-of-the-art algorithms.

In recent years, regression forests [4] have proven to be a versatile tool for solving challenging computer vision tasks efficiently. In this domain, regression forests learn a mapping from local image or depth patches to a probability over the parameter space, *e.g.*, the 2D position or the 3D orientation of the head. While related Hough forests [16] detect objects in 2D images efficiently, real-time methods for pose estimation [12, 17] rely on depth data and an abundance of synthetic training data.

In this work, we present a method based on regression forests that detects 2D facial feature points in real-time as exemplified by Fig. 1. Since regression forests learn the spatial relations between image patches and facial features from the complete training set and average the spatial distributions over all trees in the forest, the forests tend to introduce a bias to the mean face. This is very problematic for facial feature detection since subtle deformations affect only the image appearance in the neighborhood of a specific feature point. In order to steer the impact of patches close to a facial feature, which adapt better to local deformations but are more sensitive to occlusions, and more distant patches that favor the mean face, we introduce an objective function that allows to find a good trade-off.

Another contribution of this work is the introduction of conditional regression forests. In general, regression forests aim to learn the probability over the parameter space given a face image from the entire training set, where each tree is trained on a randomly sub-sampled training set to avoid over-fitting. Conditional regression forests aim to learn several conditional probabilities over the parameter space instead. The motivation is that conditional probabilities are easier to learn since the trees do not have to deal with all facial variations in appearance and shape. Since some vari-

ations depend on global properties of the face like the head pose, we can learn the trees conditional to the global face properties. During training, we also learn the probability of the global properties to model the full probability over the parameter space. During testing, a set of pre-trained conditional regression trees is selected based on the estimated probability of the global properties as illustrated in Fig. 2. For instance, having trained regression trees conditional to various head poses, the probability of the head pose is estimated from the image and the corresponding trees are selected to predict the facial features. In this way, the trees that are selected for detecting facial features might vary from image to image.

In our experiments, we demonstrate the benefit of conditional regression forests and evaluate the method on the challenging Labeled Faces in the Wild [20] database, achieving close-to-human accuracy while processing images in real-time.

2. Related Work

Facial feature detection from 2D images is a well-studied problem, especially as a preprocessing step for face recognition. Earlier works can be classified into two categories, depending on whether they use holistic or local features.

Holistic methods, *e.g.*, Active Appearance Models [2, 5, 7], use the texture over the whole face region to fit a linear generative model to a test image. Such algorithms suffer from lighting changes, modeling complexity, and a bias towards the average face. Moreover, these methods perform poorly on unseen identities [18] and deal poorly with low-resolution images.

In recent years, there has been a shift towards methods based on independent local detectors. Such detectors are discriminative models of image patches centered around the facial landmarks, often combined by enforcing a prior over their joint position. Vukadinovic and Pantic [31] train independent GentleBoost detectors for 20 facial points using Gabor filters' responses. Even though each point is localized within a limited face region, the lack of a global shape model can lead to non-plausible facial configurations. More in general, local detectors are ambiguous: the limited support region can not cope with the large appearance variations present in the training samples. To improve accuracy and reduce the influence of wrong detections, global information about the face configuration is commonly used.

In the seminal work of [6], Active Shape Models use a linear Point Distribution Model (PDM) constructed from aligned training shapes, driven to fit a new image thanks to simple models of the appearance along profiles centered on each landmark. Some extensions to the ASM algorithm were proposed, mainly focusing on improving the local detectors. For example, Constrained Local Models [9] use PCA to model the landmarks' appearance, while Boosted

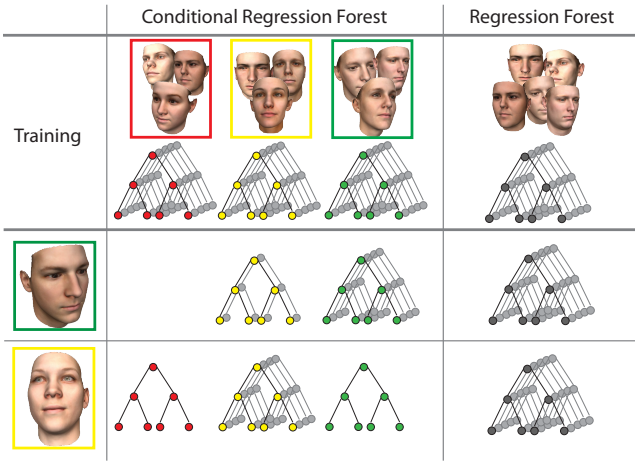


Figure 2. While a regression forest is trained on the entire training set and applied to all test images, a conditional regression forest consists of multiple forests that are trained on a subset of the training data illustrated by the head poses (colored red, yellow, green). When testing on an image (illustrated by the two faces at the bottom), the head pose is predicted and trees of the various conditional forests (red, yellow, green) are selected to estimate the facial feature points.

Regression Active Shape Models [10] use boosting to predict a new location for each point, given the patch around the current position.

Among the methods focusing on a more robust global shape prior, Everingham et al. [11] model the face configuration using pictorial structures [14], a hierarchical version of which was used in [25]. Valstar et al. [29] combine SVM regression for estimating the feature points' location with conditional Markov random fields to keep the estimates globally consistent. They also take advantage of facial feature points whose position is less sensitive to facial expressions; they thus start by localizing such stable points first and then find the additional points after a registration step. The whole process takes around 50 seconds per image. Very recently, Amberg and Vetter [1] proposed to run detectors over the whole image and then find the optimal set of detections using Branch & Bound; however, they only show results for high-quality images and need over one second to process one image.

The recent work of Belhumeur et al. [3] proposed a Bayesian model combining the outputs of the local detectors with a consensus of non-parametric global models for part locations. Their algorithm is the most accurate approach up-to-date in the literature, capable of precise localizations even in uncontrolled image conditions, like the ones present in the Labeled Face Parts in the Wild [3] dataset. However, the Labeled Face Parts in the Wild dataset contains images of higher resolution and quality compared to the LFW database [20] used for our experiments; more-

over, the reported processing time is around one second per feature point, ruling out any real-time application.

While there exist methods for detecting facial feature points accurately, there is a lack of reliable methods that achieve real-time performance.

3. Facial Feature Localization using Random Regression Forests

Random forests [4] have recently become a popular approach in computer vision. They have been used for a large number of classification [19, 26, 27] and regression [8, 12, 13, 15, 17] tasks. In this section, we outline the training and testing of a random regression forest for facial feature detection in 2D images. In Section 4, we introduce the concept of conditional regression forests. [1]

3.1. Training

Each tree T in the forest $\mathcal{T} = \{T_t\}$ is built from a different, randomly selected, set of training images. From each image, we randomly extract a set of square patches $\{\mathcal{P}_i = (\mathcal{I}_i, \mathcal{D}_i)\}$, where $\mathcal{I}_i$ represents the appearance and $\mathcal{D}_i$ represents the set of offsets to each facial feature point.

In our case, the patch appearance $\mathcal{I}_i$ is defined by multiple channels $\mathcal{I}_i = (I_i^1, I_i^2, ..., I_i^C)$. The first two channels contain the gray values of the raw input image and the normalized gray values to compensate for illumination changes. The additional channels represent a Gabor filter bank with eight different rotations and four different phase shifts. The set of offsets $\mathcal{D}_i = (\boldsymbol{d}_i^1, \boldsymbol{d}_i^2, \boldsymbol{d}_i^3, \ldots, \boldsymbol{d}_i^N)$ contains N 2D displacement vectors from the centroid of the patch to each of the N facial feature points.

We define a simple patch comparison feature, similar to [12, 21, 26]:

$$f_\theta(P) = \frac{1}{|R_1|} \sum_{\boldsymbol{q} \in R_1} I^a(\boldsymbol{q}) - \frac{1}{|R_2|} \sum_{\boldsymbol{q} \in R_2} I^a(\boldsymbol{q}), \quad (1)$$

where the parameters $\theta = (R_1, R_2, a)$ describe two rectangles R_1 and R_2 within the patch boundaries, and the selected appearance channel $a \in \{1, 2, \ldots, C\}$.

The training follows the random forest framework proposed by Breiman [4]:

1. Generate a pool of splitting candidates $\phi = (\theta, \tau)$.
2. Divide the set of patches $\mathcal{P}$ into two subsets $\mathcal{P}_L$ and $\mathcal{P}_R$ for each ϕ.

$$\mathcal{P}_L(\phi) = \{\mathcal{P} | f_\theta(P) < \tau\} \quad (2)$$

$$\mathcal{P}_R(\phi) = \mathcal{P} \setminus \mathcal{P}_L(\phi) \quad (3)$$

3. Select the splitting candidate ϕ which maximizes the evaluation function Information Gain (IG):

$$\phi^* = \arg\max_\phi IG(\phi), \quad (4)$$

$$IG(\phi) = \mathcal{H}(\mathcal{P}) - \sum_{S \in \{L,R\}} \frac{|\mathcal{P}_S(\phi)|}{|\mathcal{P}|} \mathcal{H}(\mathcal{P}_S(\phi)), \quad (5)$$

where $\mathcal{H}(\mathcal{P})$ is the defined class uncertainty measure, which will be described for our case in (6). Selecting a certain split amounts to adding a binary decision node to the tree.

4. Create leaf l when a maximum depth is reached or the information gain $IG(\phi)$ is below a predefined threshold. Otherwise continue recursively for the two subsets $\mathcal{P}_L(\phi)$ and $\mathcal{P}_R(\phi)$ at the first step.

To build our forest for facial feature detection, we use the Labeled Faces in the Wild [20] dataset. Each of the 13000 images is annotated with the coordinates of 10 facial feature points shown in Fig 1. Using the bounding box resulting from a face detection algorithm [30], all faces are rescaled to a common size. To make sure that all facial feature points are located inside the bounding box, we enlarged the box by 30%. We then sample three quaters of the training patches from inside the bounding box, and the remaining quater from the rest of the image, outside the face bounding box.

For the problem at hand, we need trees able to cast precise votes concerning the fiducial locations. Therefore, we evaluate the goodness of a split using (5), seeking to maximize the discriminative power of the tree. By maximizing this function, the class uncertainty for a split is minimized. The class uncertainty measurement is defined as:

$$\mathcal{H}(\mathcal{P}) = -\sum_{n=1}^{N} \frac{\sum_i p(c_n|\mathcal{P}_i)}{|\mathcal{P}|} \log\left(\frac{\sum_i p(c_n|\mathcal{P}_i)}{|\mathcal{P}|}\right), \quad (6)$$

$$p(c_n|\mathcal{P}_i) \propto \exp\left(-\frac{|\boldsymbol{d}_i^n|}{\lambda}\right), \quad (7)$$

where $p(c_n|\mathcal{P}_i)$ indicates the probability that the patch $\mathcal{P}_i$ belongs to the feature point n. The class affiliation is based on the distance to the facial feature point. While $p(c_n|\mathcal{P}_i) = 1$ for a patch $\mathcal{P}_i$ at the position of the n-th facial feature, it goes to zero for patches that are far away from the feature point. The factor λ controls the steepness of this function. In our experiments, we use $\lambda = 0.125$. The measure avoids a hard class assignment of the patches that is difficult to define for facial feature points. In contrast to a regression objective as used in [8] and [12], it is faster to compute and also performed slightly better in our experiments.

[1] Independent of this work, a similar concept has been applied in the context of human pose estimation [28]

When creating a leaf l, the distribution over the relative offsets to each facial feature point is stored. While modeling the distribution in a non-parametric manner using a Parzen estimate over the offsets of all patches reaching the leaf at training time as in [15] does not impose any assumption on the type of distribution, it prevents real-time performance for larger training sets. Therefore, we simplify the distribution over the offset by a multivariate Gaussian as in [8, 12]:

$$p(\boldsymbol{d}^n|l) = \mathcal{N}(\boldsymbol{d}^n; \overline{\boldsymbol{d}_l^n}, \boldsymbol{\Sigma}_l^n), \tag{8}$$

where $\overline{\boldsymbol{d}_l^n}$ and $\boldsymbol{\Sigma}_l^n$ are the mean and covariance matrix of the offsets of the nth facial feature point.

While (8) models the probability for a patch $\mathcal{P}$ ending in the leaf l of a single tree, the probability of the forest is obtained by averaging over all trees [4]:

$$p(\boldsymbol{d}^n|\mathcal{P}) = \frac{1}{T}\sum_t p(\boldsymbol{d}^n|l_t(\mathcal{P})), \tag{9}$$

where l_t is the corresponding leaf for the tree T_t.

To each leaf, we also assign a confidence weight defined by $w_l^n = \frac{1}{trace(\boldsymbol{\Sigma}_l^n)}$. The probability that the leaf l is a positive leaf for the fiducial n is defined by averaging the class certainty of each patch that reached the leaf:

$$p\left(c_n|l\right) = \frac{\sum_i p\left(c_n|\mathcal{P}_i\right)}{|\mathcal{P}|}. \tag{10}$$

3.2. Testing

We initially run a face detection algorithm [30] to find the position and the size of the face. After enlarging the bounding box of the face and rescaling the face image to a common size, we densely sample patches $\mathcal{P}_i(\boldsymbol{y_i})$ inside the bounding box, where $\boldsymbol{y}_i$ is the pixel location of the patch $\mathcal{P}_i$.

Each patch is then fed to all the trees in the random forest. At each node of a tree, the patches are evaluated according to the stored binary test and passed either to the right or left child until a leaf node is reached. The binary tests inside the trees are speeded up through the use of integral images, which greatly reduce the amount of time needed to sum the values in the subregions R_1 and R_2. By passing all the sampled patches down all the trees in the forests for facial feature detection, each patch $\mathcal{P}_i$ ends in a set of leafs $\mathcal{L}_i$.

Given a Gaussian Kernel K and the bandwidth parameter h, the density estimator for facial feature point n at pixel location $\boldsymbol{x}^n$ can be written as:

$$f\left(\boldsymbol{x}^n\right) \propto \sum_i \sum_{l\in\mathcal{L}_i} w_l^n K\left(\frac{\boldsymbol{x}^n - \left(\boldsymbol{y}_i + \overline{\boldsymbol{d}_l^n}\right)}{h}\right) \phi_n\left(l\right), \tag{11}$$

$$\phi_n\left(l\right) = \begin{cases} 1 & p\left(c_n|l\right) \geq \alpha, \\ 0 & \text{otherwise}, \end{cases} \tag{12}$$

where w_l^n is the confidence weight of the leaf l. The factor ϕ_n avoids a bias towards an average face configuration. In order to reduce the influence of votes coming from other parts of the face and to improve the efficiency, we consider only leafs with a class-affiliation higher then α. Long distance votes can give a robust estimation for the overall positions of the facial parts, but the accuracy could be low. For example, patches around the eyes provide a robust estimation of the mouth area, but they carry little information content about the exact location of the mouth corners. The facial feature points are then obtained by performing mean-shift for each point n.

4. Conditional Regression Forest

While a regression forest aims to model the probability $p(\boldsymbol{d}^n|\mathcal{P})$ (9) given an image patch $\mathcal{P}$, a conditional regression forest models the conditional probability $p(\boldsymbol{d}^n|\omega, \mathcal{P})$ and estimates (9) by

$$p(\boldsymbol{d}^n|\mathcal{P}) = \int p(\boldsymbol{d}^n|\omega, \mathcal{P}) p(\omega|\mathcal{P})\, \mathrm{d}\omega, \tag{13}$$

where ω is an auxiliary parameter that can be estimated from the image. In our case, ω corresponds to the head pose that can be estimated as described in Section 4.1.

In order to learn $p(\boldsymbol{d}^n|\omega, \mathcal{P})$, the training set is split into subsets, where the space of the parameter ω is discretized into disjoint sets Ω_i. Hence, (13) becomes

$$p(\boldsymbol{d}^n|\mathcal{P}) = \sum_i \left(p(\boldsymbol{d}^n|\Omega_i, \mathcal{P}) \int_{\omega\in\Omega_i} p(\omega|\mathcal{P})\, \mathrm{d}\omega \right). \tag{14}$$

The conditional probability $p(\boldsymbol{d}^n|\Omega_i, \mathcal{P})$ can be learned by training a full regression forest $\mathcal{T}(\Omega_i)$ as in Section 3 on each of the training subsets Ω_i. Similarly, the probability $p(\omega|\mathcal{P})$ can be learned by a regression forest on the full training set Ω as described in Section 4.1.

While regression forests average the probabilities over all trees T_t (9), we select T trees from the conditional regression forests $\mathcal{T}(\Omega_i)$ based on the estimated probability $p(\omega|\mathcal{P})$. To this end,

$$p(\boldsymbol{d}^n|\mathcal{P}) = \frac{1}{T}\sum_i \sum_{t=1}^{k_i} p(\boldsymbol{d}^n|l_{t,\Omega_i}(\mathcal{P})), \tag{15}$$

where l_{t,Ω_i} is the corresponding leaf for patch $\mathcal{P}$ of the tree $T_t \in \mathcal{T}(\Omega_i)$. The discrete values k_i are computed such that $\sum_i k_i = T$ and

$$k_i \approx T \cdot \int_{\omega\in\Omega_i} p(\omega|\mathcal{P})\, \mathrm{d}\omega. \tag{16}$$

4.1. Head Pose Estimation

To obtain the head pose, we train a regression forest similar to [13, 19]. To this end, we quantize the training data into 5 subsets that correspond to 'left profile', 'left', 'front', 'right', and 'right profile' faces since it is diffcult to obtain continuous ground truth head pose data from 2D images. As for the facial feature detection, we rescale the faces based on the face detection result. Since the face bounding box is not always perfect and sometimes contains many background pixels, we train trees which are able to classify patches that belong to the face and predict the head pose at the same time. To this end, we use

$$\mathcal{H}_{pose}\left(\mathcal{P}\right) = -\sum_{c} p\left(c|\mathcal{P}\right) \log\left(p\left(c|\mathcal{P}\right)\right) \tag{17}$$

as evaluation function (5). At each node, we randomly choose if c corresponds to the fore- and background labels or to the labels of the head pose class affiliation.

After replacing the head pose labels by real world angles $\omega \in \{-90, -45, 0, +45, +90\}$ representing the yaw angle, we store the multivariate Gaussian distribution

$$p(\omega|l) = \mathcal{N}(\omega; \overline{\omega_l}, \mathbf{\Sigma}_l) \tag{18}$$

in each leaf. In our case, we achieved more robust estimates by converting the discrete labels into continuous values than estimating the discrete head pose class labels directly.

5. Experiments

Dataset Many face databases annotated with facial features exist. The most common ones are BioID[2] (annotations: FGnet project[3]), AR [22], and FERET [23]. All were either acquired under controlled lighting conditions or contain only frontal faces, *i.e.*, none of the above can be considered realistic for many applications.

Exceptions exist, like the Labeled Face Parts in the Wild (LFPW) [3] and the Labeled Faces in the Wild (LFW) [20] databases, containing large variations in the imaging conditions. While LFPW is annotated with facial point locations, only a subset of about 1500 images is made available; moreover, LFPW contains better quality images compared to LFW. The LFW database contains facial images of 5749 individuals, 1680 of which have more than one image in the database. The images have been collected 'in the wild' and vary in pose, lighting conditions, resolution, quality, expression, gender, race, occlusion, and make-up. We annotated 13,233 faces taken from LFW database with the location of 10 facial feature points shown in Fig. 3. We used Amazon Mechanical Turk, labeling each fiducial point at least three times and taking the mean of the annotations as ground truth.

[2] www.bioid.com

[3] www-prima.inrialpes.fr/FGnet/html/benchmarks.html

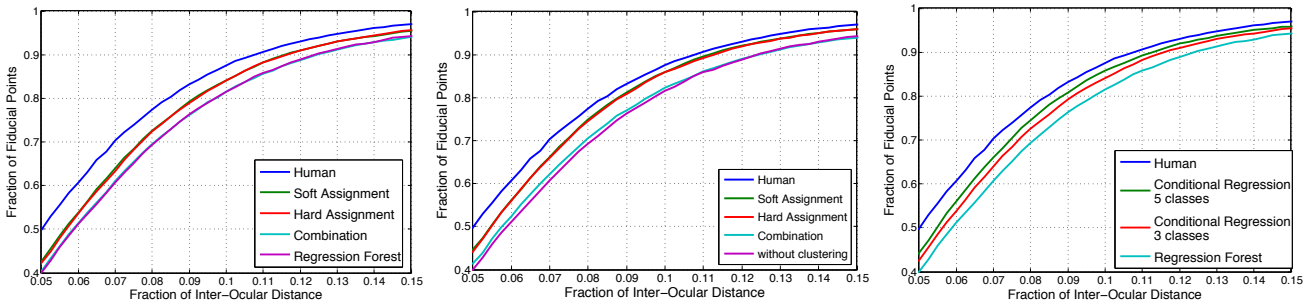
Evaluation As in previous work, we measure the localization error as a fraction of the inter-ocular distance, a measure invariant to the actual size of the images. We declare a point correctly detected if the pixel error is below 0.1 inter-ocular distance, a very stringent measure, as exemplified by Fig. 3. We also compare to the performance of human annotators, measured as in [3] by calculating the average error of a MTurk user in comparison to the mean of the other users.

Figure 3. Two images of the LFW dataset annotated with 10 facial feature points. The white circles show the error tolerance (10% of the inter-ocular distance).

Training For training the regression trees, we fixed some parameters on the basis of empirical observations, *e.g.*, the trees have a maximum depth of 20 and at each node we randomly generate 2500 splitting candidates and 25 thresholds. Each tree is grown based on a randomly selected subset of 1500 images. After running the face detection algorithm on each image, we rescale the face bounding box to 100×100 pixels and enlarge it by 30% to make sure that all facial features are enclosed. We then extract fixed size (20×20 pixels) patches from each training image, 150 from within the face and 50 from outside.

Testing Test-time parameters include the number of mean-shift iterations, the bandwidth of the mean-shift kernel h (11), and the α (12) which limits the impact of distant votes. Such parameters are automatically estimated during training from a validation set generated from the training data by randomly extracting 100 patches from every training image. This means that it is possible but improbable that the evaluation set contains patches which are also part of the training set. On these out of bag data, we then perform a grid search to determine the best parameters.

The most important parameter turns out to be α (12). When equal to zero, all patches contribute to the mean shift, while only patches in a small neighborhood are taken into account when α is close to one. Fig. 4 shows the impact of α on the detection accuracy: Without the thresholding, the detector tends towards the mean face and the accuracy is below 70%; by removing distant votes, the detection relies more on local patches and the performance increases significantly over 80% for α around 0.3. When the neighborhood becomes very small ($\alpha > 0.75$), the approach fails due to



(a) (b) (c)

Figure 5. Performance of conditional regression forests compared to human performance and to a standard regression forest (without clustering) using 3 clusters (a) and 5 clusters (b). Combining all trained conditional regression trees does not outperform the regression forest. Selecting the trees based on (16) improves the accuracy (soft/hard assignment) in particular for 5 clusters where the performance is close to human performance. (c) At a 0.1 inter-ocular distance threshold, the accuracy increases from 81.57% to 86.10% (5 clusters) where humans achieve 87.5%.

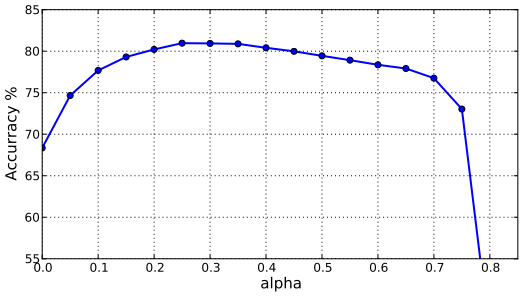


Figure 4. Average accuracy of the fiducial localization. Different values for the parameter α are evaluated.

the small number of votes.

Head Pose For the head pose estimation, we use the same settings as for the facial feature localization. Each image in the dataset is manually annotated with one out of the five head pose labels (profile left, left, front, right, profile right). We train a forest of 10 trees with a maximal depth of 10. For estimating the head pose, we process only 1 out of 25 image patches. Our final head pose estimator reaches an accuracy of 72.15%, computed over a ten-fold cross validation. Examples of the estimated head pose are shown in Fig. 1.

Conditional Regression Forests For evaluating the contribution of the conditional regression forests, we grouped the training data depending on the head pose into 5 (profile left, left, front, right, profile right) and 3 (left, front, right) clusters. We then trained a complete forest for each subset.

We compared three ways of selecting T trees for the conditional regression forest. While (16) selects the trees conditional to the estimated head pose probability (soft assignment), one can also select only trees for the head pose with the highest probability (hard assignment). An additional naïve approach (combination) randomly selects the trees without taking the estimated pose into account. Fig. 5 shows the results: While soft and hard assignment perform similar and both outperform the standard regression forest, the naïve approach does not improve the regression forest. Using 5 instead of 3 clusters improves the performance of the conditional regression forest by an additional 2%.

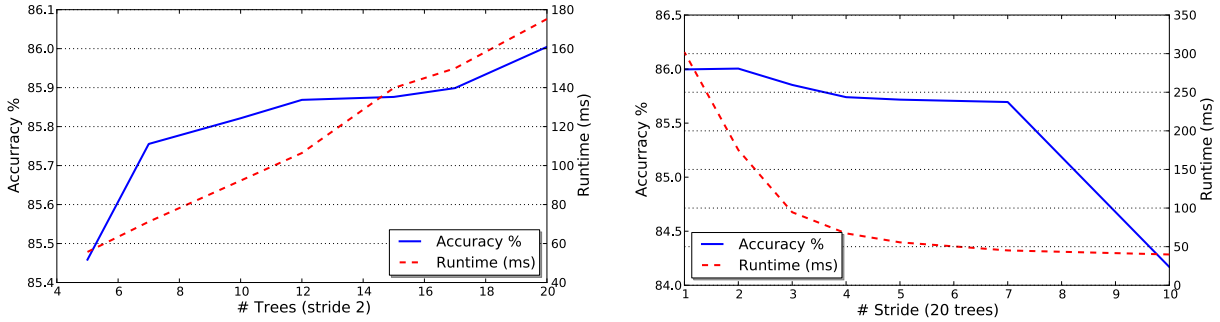
Regression forests provide two convenient parameters for finding the right trade-off between runtime and accuracy, namely the number of trees to be loaded and the sampling stride, *i.e.*, the distance between patches sampled at test time. As shown in Fig. 6, a higher number of trees and a lower stride improve the accuracy at the cost of a higher average computation time[4]. The stride parameter is crucial when real-time performance is needed: a stride greater or equal to 3 already allows for over 10 fps (*i.e.*, below 100 ms for one frame) at a marginal loss in accuracy. Notice that the rescaling and normalizing of the input image and feature extraction already takes about 19 ms and that the head pose estimation needs another 14 ms. A video demonstrating the real-time performance is part of the supplementary material.

In order to accurately measure the performance of our system, we performed a ten-fold cross validation experiment. We compare our results to two state-of-the-art methods in Fig. 7. We used the publicly available facial features detectors of Valstar et al. [29] and Everingham et al. [11] on our dataset. Our method clearly outperforms both competitors with respect to accuracy and runtime, but we have to point out that the other methods were not trained on the same dataset. For further comparisons, we have made the source code of our approach and the annotations of the dataset publicly available[5].

The error for each facial feature point is given in Fig. 7

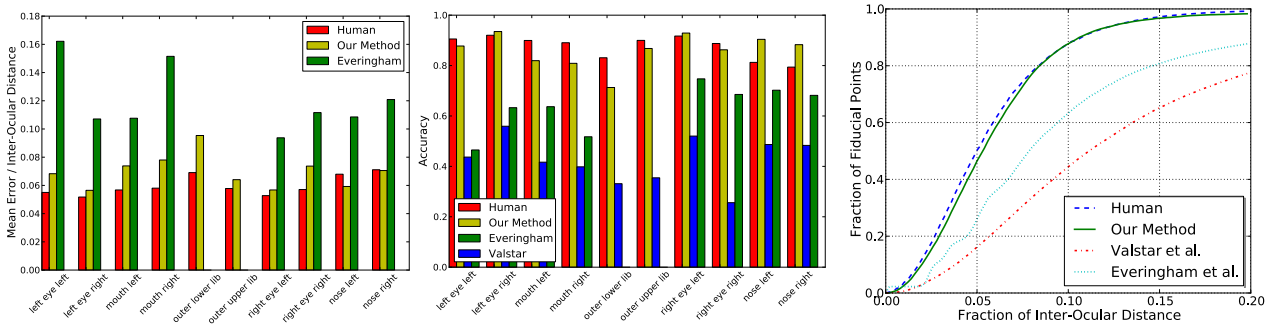
[4]Measured on 1000 randomly selected images using Intel Core i7 3.06GHz with 4 cores (multi-threaded).

[5]http://www.vision.ee.ethz.ch/~mdantone



(a) (b)

Figure 6. Trade-off between runtime and accuracy. (a) Number of trees. (b) Stride. A stride greater than 3 allows 10 fps.



(a) (b) (c)

Figure 7. Comparison to other methods. (a) Mean error. (b) Accuracy. (c) Accuracy plotted against the success thresholds.

facial feature	Cond. RF accuracy[%]	mean error	Everingham mean error
left eye left	87.7	0.0682	0.1621
left eye right	93.5	0.0565	0.1070
right eye left	92.9	0.0567	0.0937
right eye left	86.2	0.0736	0.1116
mouth left	81.9	0.0738	0.1076
mouth right	80.8	0.0780	0.1514
nose strip left	90.4	0.0592	0.1085
nose strip right	88.2	0.0705	0.1208
upper outer lib	86.7	0.0640	-
lower outer lib	71.5	0.0953	-

Table 1. Detection accuracy for all facial feature points.

and Table 1. For the inner corners of the eyes and for the two nose strips, we achieve an accuracy that is comparable to human performance. The most difficult point to detect is the lower lip. Fig. 8 shows some qualitative results. In particular, the last row shows some failure cases due to occlusions or a head pose that is not well represented in the training data.

6. Conclusions

We have presented a real-time algorithm for facial feature detection based on the novel concept of conditional regression forests. Such ensembles of regression trees estimate the position of several facial landmarks conditional to the probability of some global face properties. In this work, we have demonstrated the benefits of conditional regression forests by modeling the appearance and location of facial feature points conditional to the head pose. The proposed method achieves an accuracy comparable to the performance of human annotators on a large, challenging database of faces captured "in the wild". In our future work, we intend to model other properties like sunglasses or facial hair that still cause some problems as well. We also believe that the concept of conditional regression forests might be relevant for other computer vision applications.

Acknowledgements The authors acknowledge financial support from the SNF project Vision-supported Speech-based Human Machine Interaction (200021-130224) and the EC projects RADHAR (FP7-ICT-248873) and TANGO (FP7-ICT-249858).

References

[1] T. Amberg, B. Vetter. Optimal landmark detection using shape models and branch and bound. In *ICCV*, 2011.

[2] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 56(1):221 – 255, 2004.

[3] P. Belhumeur, D. Jacobs, D. Kriegman, and N. Kumar. Localizing parts of faces using a consensus of exemplars. In *CVPR*, 2011.

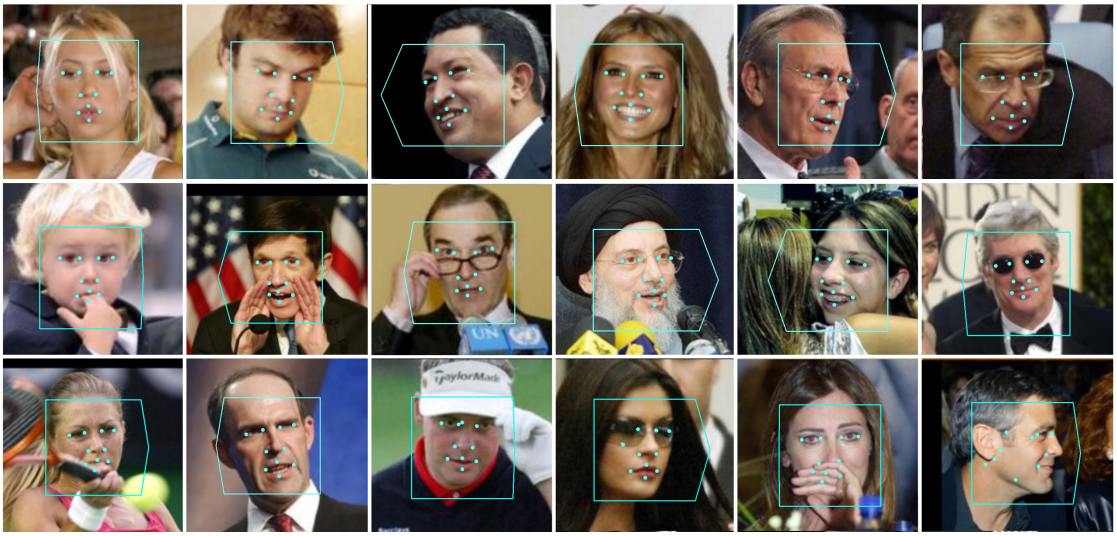


Figure 8. Qualitative results on some images from the Labeled Faces in the Wild dataset. The last row shows different error cases.

[4] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[5] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *TPAMI*, 23:681–685, 2001.

[6] T. Cootes and C. Taylor. Active shape models - 'smart snakes'. In *BMVC*, 1992.

[7] T. Cootes, K. Walker, and C. Taylor. View-Based Active Appearance Models. In *Image and Vision Computing*, pages 227–232, 2002.

[8] A. Criminisi, J. Shotton, D. Robertson, and E. Konukoglu. Regression forests for efficient anatomy detection and localization in ct studies. In *Medical Computer Vision Workshop*, 2010.

[9] D. Cristinacce and T. Cootes. Feature detection and tracking with constrained local models. In *BMVC*, 2006.

[10] D. Cristinacce and T. Cootes. Boosted regression active shape models. In *BMVC*, 2007.

[11] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy - automatic naming of characters in tv video. In *BMVC*, 2006.

[12] G. Fanelli, J. Gall, and L. Van Gool. Real time head pose estimation with random regression forests. In *CVPR*, 2011.

[13] G. Fanelli, T. Weise, J. Gall, and L. Van Gool. Real time head pose estimation from consumer depth cameras. *DAGM*, 2011.

[14] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61, 2005.

[15] J. Gall and V. Lempitsky. Class-specific hough forests for object detection. In *CVPR*, pages 1022–1029, 2009.

[16] J. Gall, A. Yao, N. Razavi, L. Van Gool, and V. Lempitsky. Hough forests for object detection, tracking, and action recognition. *TPAMI*, 2011.

[17] R. Girshick, J. Shotton, P. Kohli, A. Criminisi, and A. Fitzgibbon. Efficient regression of general-activity human poses from depth images. *ICCV*, 2011.

[18] R. Gross, I. Matthews, and S. Baker. Generic vs. person specific active appearance models. *Image and Vision Computing*, 23:1080 – 2093, 2005.

[19] C. Huang, X. Ding, and C. Fang. Head pose estimation based on random forests for multiclass classification. In *ICPR*, pages 934–937, 2010.

[20] G. Huang, M. Ramesh, T. Berg, and E. Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Technical report, University of Massachusetts, Amherst, 2007.

[21] V. Lepetit, P. Lagger, and P. Fua. Randomized trees for real-time keypoint recognition. In *CVPR*, 2005.

[22] A. Martinez. The ar face database. *CVC Technical Report*, 24, 1998.

[23] P. Phillips, H. Wechsler, J. Huang, and P. Rauss. The feret database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing*, 16(5):295–306, 1998.

[24] V. Rapp, T. Senechal, K. Bailly, and L. Prevost. Multiple kernel learning svm and statistical validation for facial landmark detection. In *FG*, pages 265–271, 2011.

[25] G. Roig, X. Boix, F. De la Torre, J. Serrat, and C. Vilella. Hierarchical crf with product label spaces for parts-based models. In *FG*, pages 657–664, 2011.

[26] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, 2011.

[27] J. Shotton, M. Johnson, and R. Cipolla. Semantic texton forests for image categorization and segmentation. In *CVPR*, 2008.

[28] M. Sun, P. Kohli, and J. Shotton. Conditional Regression Forests for Human Pose Estimation. In *CVPR*, 2012.

[29] M. Valstar, B. Martinez, X. Binefa, and M. Pantic. Facial point detection using boosted regression and graph models. In *CVPR*, pages 2729–2736, 2010.

[30] P. Viola and M. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

[31] D. Vukadinovic and M. Pantic. Fully automatic facial feature point detection using gabor feature based boosted classifiers. In *IEEE Int. Conf. on Systems, Man and Cybernetics*, pages 1692–1698, 2005.